Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method of initializing an electronic device, comprising the steps of:

in a system in which instruction caching is unsupported when executing from a first memory, beginning execution of initialization code [[in a]] from the first memory, which is organized as a plurality of pages, to copy a first portion of the initialization code from the first memory into a second memory;

~~continuing execution of~~ while executing the initialization code [[in]] from the first memory, performing a first instruction to software-enable instruction caching, wherein the first instruction is written at the end of a first page of the first memory and the subsequent page following the first page is un-initialized, such that hardware on the electronic device will automatically and temporarily disable instruction caching for the second page that is un-initialized; and

after execution of the first instruction, executing at least some of the first portion of initialization code ~~copied into~~ from the second memory.

2. (Currently amended) The method of Claim 1, wherein the step of executing at least some of the first portion of initialization code from the second memory comprises copying a second portion of the initialization code from the first memory into a third memory.

3. (Currently amended) The method of Claim 2, further comprising the step of executing the second portion of initialization code ~~copied into the second~~ from the third memory.

4. (Cancelled)

5. (Currently amended) The method of Claim 1, wherein instruction caching is enabled ~~during~~ for the step of executing at least some of the first portion of initialization code from the second memory.

6. (Currently amended) The method of Claim 3, wherein instruction caching is enabled ~~during~~ for the step of executing the second portion of initialization code from the third memory.

7. (Original) The method of Claim 1, wherein the first memory is at least one non-volatile storage device.

8. (Original) The method of Claim 2, wherein the second memory is cache memory and the third memory is system memory.

9-12. (Cancelled)

13. (Currently amended) An apparatus comprising:

a processor,

a first memory, which is organized as a plurality of pages, and a second memory, wherein the processor does not support instruction caching when executing from the first memory; ~~comprises~~

initialization code including a first portion and a second portion, the ~~first portion~~ initialization code having

first instructions, executable from the first memory, for copying the first portion into the second memory,

second instructions, executable from the first memory, for enabling instruction caching for the processor, wherein the second instructions are stored at the end of a first page and the page following the second instructions is un-initialized, such that the processor automatically and temporarily disables instruction caching for the second page that is un-initialized,

third instructions, executable from the first memory, for transferring execution control to execute from the second memory, and

fourth instructions, executable from the second memory, for copying the second portion into a third memory.

14-15. (Cancelled)

16. (Original) The apparatus of Claim 13, wherein the first memory is at least one non-volatile storage device.

17. (Original) The apparatus of Claim 16, wherein the second memory is cache memory and the third memory is system memory.

18-23. (Cancelled)

24. (Currently amended) A computer readable tangible medium containing a computer program product ~~on a computer readable medium~~, the computer program product having initialization code for initializing an apparatus, the initialization code comprising:

first code, executable from a first memory of an apparatus that does not support instruction caching while executing from the first memory, for copying a first portion of the initialization code from [[a]] the first memory, which is organized as a plurality of pages, to a second memory;

second code, executable from the first memory, for software enabling instruction caching in the apparatus, wherein the second code is stored at the end of a first page of the first memory and the page following the first page is un-initialized, such that hardware on said electronic device will automatically and temporarily disable instruction caching for the second page that is un-initialized; and

third code, performed after execution of the second code, for transferring execution control to continue execution of the first portion ~~copied in~~ from the second memory with instruction caching enabled.

25. (New) A device comprising:

a non-volatile memory containing initialization code, the non-volatile memory being organized as a plurality of pages, wherein the initialization code is stored in the non-volatile memory such that a first instruction to enable instruction caching is stored at the end of a first page in the non-volatile memory and the page following the first page is un-initialized;

a cache memory;

a system memory; and

a processor, connected to the non-volatile memory, the cache memory and the system memory, wherein the processor hangs if instruction caching is attempted while executing from the first memory;

wherein the initialization code is first executed from the non-volatile memory while a first portion of the initialization code is copied to the cache memory, the first portion of the initialization code is next executed from the cache memory while a second portion of the initialization code is copied to the system memory, and the second portion of the initialization code is finally executed from the system memory;

wherein the storage location of the first instruction provides that instruction caching be software-enabled from the non-volatile memory but be hardware inhibited until execution passes from the non-volatile memory to the cache memory.

26. (New) A method of initializing an electronic device that hangs if instruction caching is attempted while executing from a non-volatile memory, the method comprising:

providing the electronic device with a cache memory; a system memory; and a non-volatile memory that contains initialization code, the non-volatile memory being organized as a plurality of pages, wherein the initialization code is stored in the non-volatile memory such that a first instruction to enable instruction caching is stored at the end of a first page in the non-volatile memory and the page following the first page is un-initialized;

initially executing the initialization code from the non-volatile memory while a first portion of the initialization code is copied to the cache memory;

executing the first portion of the initialization code from the cache memory while a second portion of the initialization code is copied to the system memory; and

executing the second portion of the initialization code from the system memory;

wherein the storage location of the first instruction provides that instruction caching be software-enabled from the non-volatile memory but be hardware inhibited until execution passes from the non-volatile memory to the cache memory.